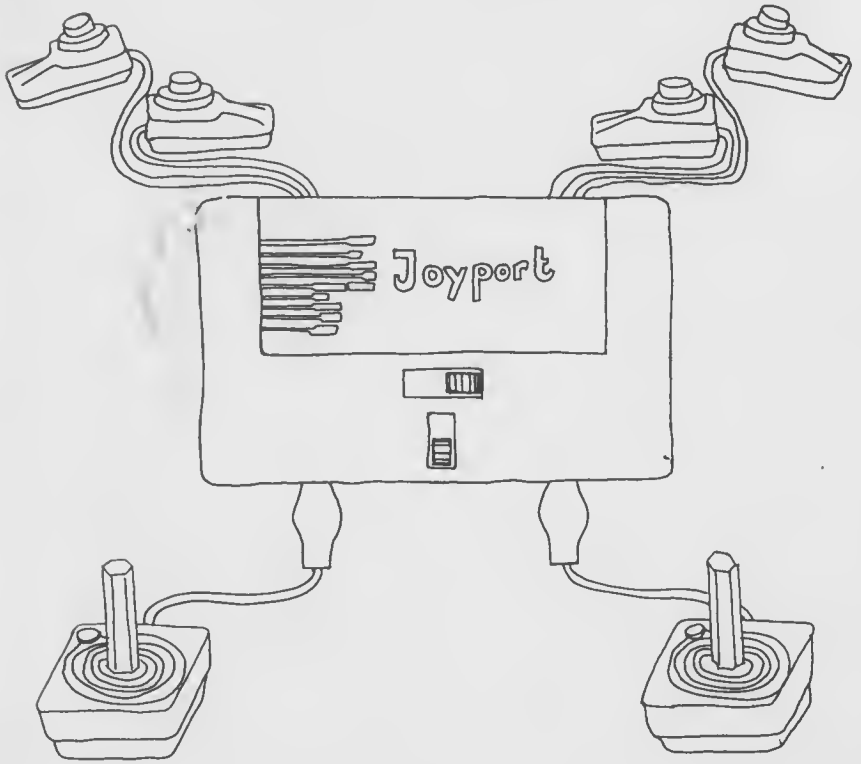
The JOYPORT

USER MANUAL

Apple II Game Port Expander



Sirius Software, Inc.
10364 Rockingham Drive, Sacramento, California 95827

The JOYPORT

USER MANUAL

Apple II Game Port Expander

Sirius Software, Inc.
10364 Rockingham Drive, Sacramento, California 95827

Table Of Contents

Introduction

The JOYPORT is the most significant new input device for the Apple computer since the keyboard. The JOYPORT provides expansion of the game paddle port to allow the use of two separate Apple devices and two Atari-type joysticks. All the connections for the Apple and Atari paddles and joysticks are easily accessible without opening the Apple case. The JOYPORT is compatible with present games and programs that use the Apple paddles. A wide variety of software is rapidly being developed to take advantage of the added features of the JOYPORT. This includes most products that Sirius Software, Inc. will be publishing as well as software from other vendors.

This manual contains complete instructions for installation of the JOYPORT, operating instructions and programming examples for interfacing all the JOYPORT functions to BASIC, Pascal and assembly language programs.

Apple Controllers:

The JOYPORT has two complete Apple game I/O sockets (16 pin DIP). Any two Apple controllers can be plugged in. Controllers include paddles, joysticks and software protection devices.

Atari Controllers:

Two Atari-type (DE-9) sockets are provided on the front of the JOYPORT. Two Atari™ or Le-Stick™ (by Datasoft) type joysticks can be plugged into these sockets.

The Controllers

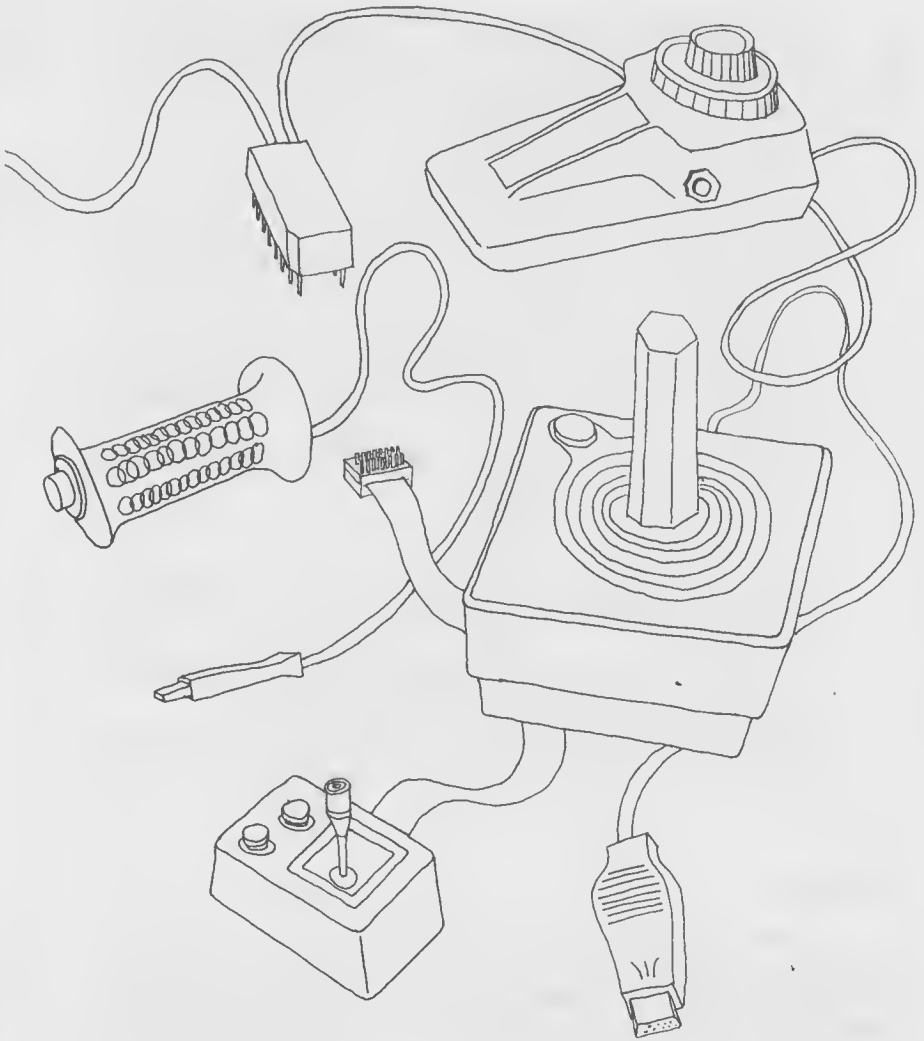


Figure 1

Please carefully follow the installation instructions given below. You can cause damage to either the Apple or the JOYPORT if you install the JOYPORT incorrectly! Always make sure you turn off the Apple before installing or removing the JOYPORT, paddles or joysticks.

1: Pull the top off your Apple II Computer.

2: Locate the GAME I/O socket on the Apple printed circuit board. It is located in the upper right hand corner of the printed circuit board when the keyboard is facing you. It is marked with white lettering just above the socket saying "GAME I/O."

3: Plug in the JOYPORT connector cable. Make sure the cable is oriented so the flat cable sticks out on the right side. The beveled edge of the connector should be facing towards the keyboard.

4: Thread the cable out of the Apple case through one of the cut outs in the back of the Apple.

5: Pop off the top of the JOYPORT box.

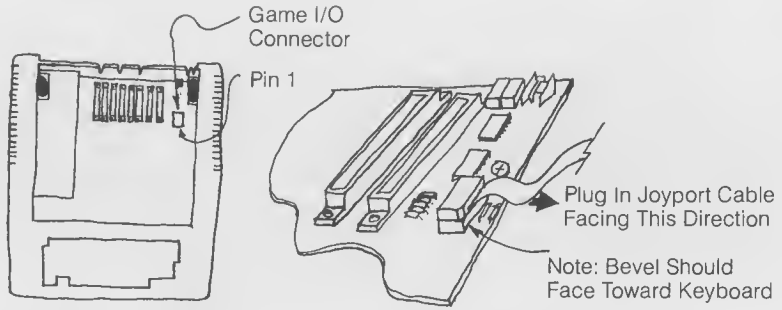
6: Plug in your game paddles and joysticks. Make sure that pin 1 of each device you plug in is facing towards the front of the JOYPORT box.

7: Run the joystick and game paddle cables out the cut outs in the back of the JOYPORT box and replace the top.

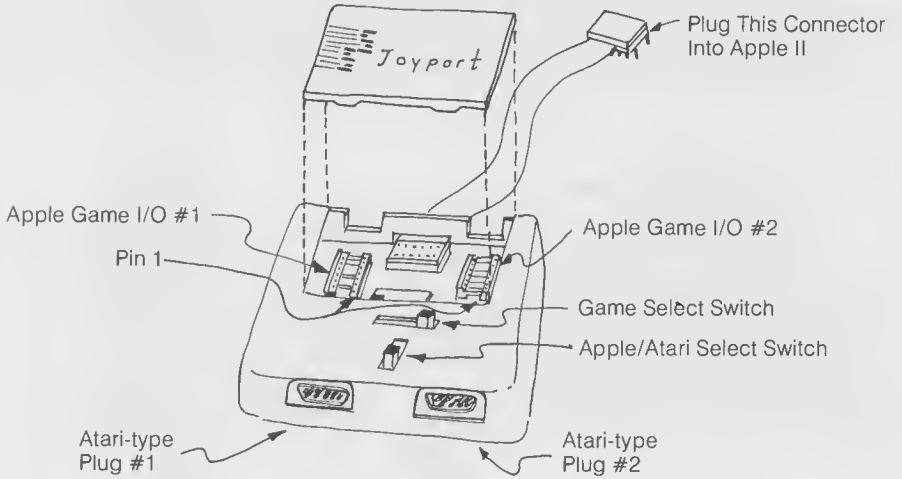
8: Plug in the Atari-type joysticks in the plugs provided on the front of the JOYPORT box.

Warning:

Make sure that the game I/O plugs are inserted so that all their pins are in the socket. It is very easy to accidently push the connectors in with 1 set of pins off. Also make sure that pin 1 of the connector (the one next to the beveled corner) is plugged into pin 1 of the socket. Refer to the drawings on the opposite page for pin 1 locations. Finally, always make sure that your Apple computer is turned off before plugging or unplugging any controllers.



Apple Top View *Apple PC Board*



Joyport Layout

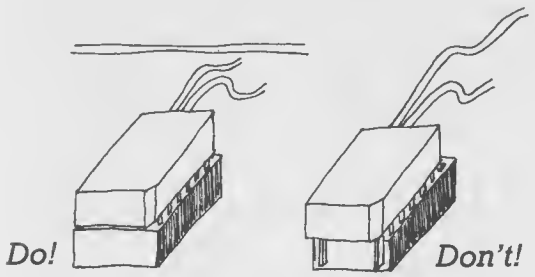


Figure 2

Using The Joyport

The two switches on the top of the JOYPORT are used to select which set of controllers the Apple will respond to. The horizontally aligned switch (the one labeled CONTROLLER SELECT SWITCH on fig. 3) switches input between the left and right sets of game paddles or Atari-type joysticks. The verticle switch (the one labeled APPLE/ATARI SELECT SWITCH on fig. 3) switches input between the Apple and Atari-type controllers.

Apple/Atari Select Switch:

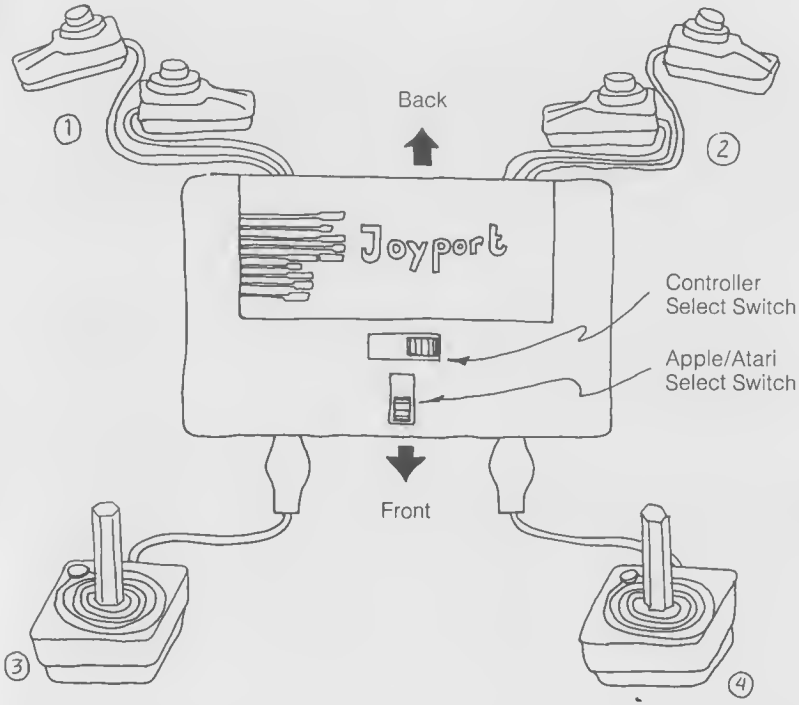
You can activate either the Apple controllers or the Atari controllers. If you wish to use the Apple controllers for a game or program then push the APPLE/ATARI SWITCH toward the back of the JOYPORT. Pulling the switch toward the front of the JOYPORT will activate the Atari controllers.

Controller Select Switch:

Once you have selected Apple or Atari controllers you can set the JOYPORT so that the right set, the left set or both sets of controllers are active by moving the CONTROLLER SELECT SWITCH. If you wish to only use the controller plugged into the left socket of the JOYPORT then move this switch to the left. Setting it to the right side will allow input only from the right set of controllers. If your software requires both the left and right sets of controllers to be active then move the switch to the middle setting.

Refer to the operating instructions supplied with the game or program you wish to run for further information about activating the special features of the JOYPORT. If you are a programmer, please refer to the following pages for a more detailed explanation of how the switch settings affect the operation of the JOYPORT.

The Joyport Set Up



Switch Settings

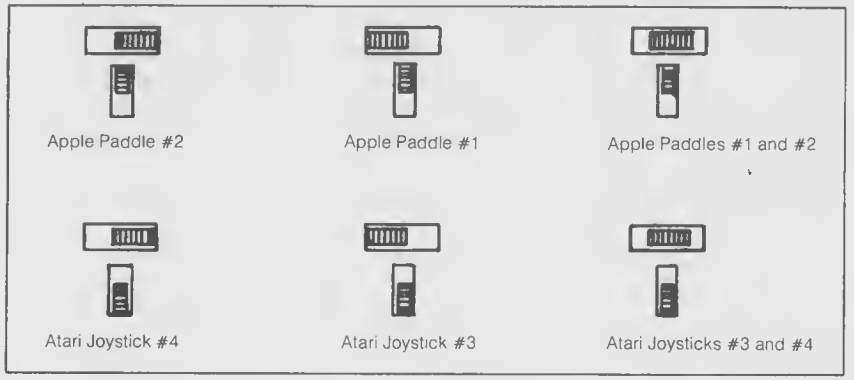


Figure 3

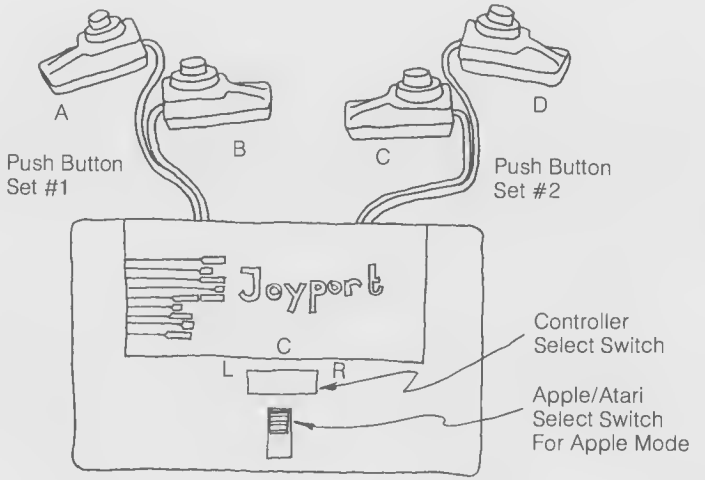
Programming With Apple Controllers

At all times, +5 Volts, ground, the strobe and annunciators are available on both the 16 pin DIP Apple sockets in the JOYPORT. The pin assignments of both these sockets are exactly the same as those for the GAME I/O socket inside the Apple II. Refer to the Apple II Reference Manual for pin assignments. When the JOYPORT is set in the Apple mode (ATARI/APPLE switch set to rear) you have two complete Apple game I/O sockets to choose from. The rear switch on the JOYPORT will select between the left, right, or both game sockets.

Pushing the CONTROLLER SELECT SWITCH to either the extreme right or left will select the corresponding socket. Access to all the controls will be the same as if the selected set of paddles were plugged directly into the Apple GAME I/O socket.

When the CONTROLLER SELECT SWITCH is placed in the middle position, all four game paddles can be read. Reading pdl(0) or pdl(1) will give the values associated with paddle 0 and paddle 1 inputs from the left socket, while paddle 0 and 1 inputs from the right are read as pdl(2) and pdl(3). Annunciator zero is used to select between pushbutton inputs from the two sides. Turning off annunciator 0 ($C058 or −16296) will allow pushbutton inputs on the left side to be read normally, while switching it on ($C059 or −16295) will allow those on the right to be read.

The table on the opposite page illustrates which paddles (A,B,C,D) will be read with various settings of both the CONTROLLER SELECT SWITCH and annunciator 0. Note that most Apple game paddles come with only two paddles wired into the connector. It is possible to wire four analog inputs into a single socket. If you have done so, then all four of the paddles that are plugged into the single socket can be read only when that socket is selected i.e., when the CONTROLLER SELECT SWITCH is in the extreme left or right. If you place the CONTROLLER SELECT SWITCH in the middle only pdl(0) and pdl(1) will be read from one socket.



Apple Mode Function Selection

	Controller Select Switch Setting	PDL (0)	PDL (1)	PDL (2)	PDL (3)	Buttons
	Left	A	B	Optional*	Optional*	Set #1
	Right	C	D	Optional*	Optional*	Set #2
CENTER	Ann. 0 On	A	B	C	D	Set #2
CENTER	Ann. 0 Off	A	B	C	D	Set #1

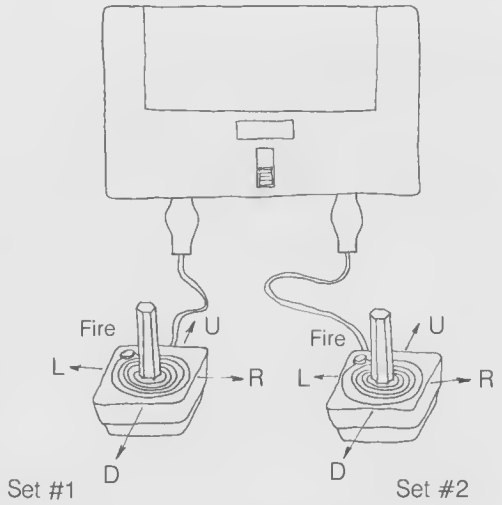
*Note: If you have 4 paddles wired into a single socket, PDL (2) and PDL (3) will be available.

Figure 4

To use Atari-style joysticks, the Apple/Atari switch must be pulled forward to the Atari setting. Just as in the Apple configuration, the CONTROLLER SELECT SWITCH will enable you to select between use of the right, left, or both joystick inputs. As explained above, with the switch in the center position, annunciator 0 low ($C058 or −16296) will select the left side, while setting it high ($C059 or −16295) will allow reading of the right side.

Internally, Atari-type joysticks have 5 pushbutton switches. Each of the four cardinal points of the main axis have one push button. The fifth button is used for the fire button. Since the Apple game I/O port has only 3 push button inputs, the JOYPORT uses annunciator 1 to selectively read either the horizontal or vertical axis button sets. Annunciator 1 low ($C05A or −16294) will select the horizontal axis, while setting it high ($C05B or −16293) will allow reading of the vertical axis.

The fire button is read from button 0 ($C061 or −16287). The other two switches are read from buttons 1 and 2 ($C062 & $C063 or −16286 & −16285). Refer to the diagram on the opposite page for the orientation of the buttons and selection of the switches using the CONTROLLER SELECTOR SWITCH and the annunciators.



Atari Mode Function Selection

Controller Select Switch Setting	Annunciator #1	Button 0 $C061 −16287	Button 1 $C062 −16286	Button 2 $C063 −16285
Left	On	Fire-1	Up-1	Down-1
	Off	Fire-1	Left-1	Right-1
Right	On	Fire-2	Up-2	Down-2
	Off	Fire-2	Left-2	Right-2

Controller Select In Middle

Annunciator #0	Annunciator #1	Button 0 $C061 −16287	Button 1 $C062 −16286	Button 2 $C063 −16285
On	On	Fire-1	Up-1	Down-1
	Off	Fire-1	Left-1	Right-1
Off	On	Fire-2	Up-2	Down-2
	Off	Fire-2	Left-2	Right-2

Figure 5

Appendix A: BASIC Listings

```
100  REM  JOYPORT TEST PROGRAM
200  POKE  - 16296,0: REM  ANNUNCIATOR 0 OFF
210  POKE  - 16294,0: REM  ANNUNCIATOR 1 OFF
300  POKE  - 16368,0: REM  CLEAR KEYBOARD STROBE
400  TEXT : HOME : REM  CLEAR SCREEN
500  HOME : PRINT "JOYPORT TEST PROGRAM": PRINT : PRINT
600  PRINT "PLUG IN JOYPORT  TO TEST AND PLUG TWO": PRINT "ATARI JOYSTICK
     S AND TWO SETS OF GAME    PADDLES TO IT."
700  PRINT "SET THE FRONT SWITCH TO THE FRONT AND   THE BACK SWITCH TO TH
     E RIGHT"
800 JT$ = "RIGHT": POKE  - 16296,0: POKE  - 16294,0: GOSUB 1300: REM   TE
     ST RIGHT ATARI STICK
900  POKE  - 16296,0: POKE  - 16294,0:JT$ = "LEFT": GOSUB 3500: REM    TE
     ST LEFT ATARI STICK
1000  GOSUB 3900: REM  READ BOTH ATARI STICKS WITH BACK SWITCH CENTERED
1100  GOSUB 4400: END : REM  PADDLE READ
1200  HOME : PRINT "OK, THAT JOYPORT WAS OK. PLUG ANOTHER   ONE IN AND PR
     ESS SPACE TO RERUN THE     TEST.": GET A$: RUN
1300  POKE  - 16368,0: REM  CLEAR KEYBOARD STROBE
1400  PRINT "PRESS SPACE WHEN READY TO START": GET A$: POKE  - 16368,0
1500  HOME : PRINT "TILT THE "JT$" ATARI JOYSTICK RIGHT"
1600  FOR X = 1 TO 1000: IF  PEEK ( - 16285) < 128 THEN 1900: REM  CHECK
     BUTTON 0
1700  NEXT X: REM  WAIT LOOP. IF NO RESPONSE THE TESTER DIED THE JOYPORT
     FAILED, THE SWITCHES WERE SET WRONG OR HELL FROZE OVER. MAKE THEM ST
     ART OVER AGAIN.
1800  GOTO 3800
1900  HOME : PRINT "OK THAT WORKS, NOW TILT THE JOYSTICK    LEFT"
2000  FOR X = 1 TO 1000: IF  PEEK ( - 16286) < 128 THEN 2300: REM BUTTON1

2100  NEXT X: REM  SEE NOTES ON 1700
2200  GOTO 3800
2300  HOME : PRINT "OK, THAT WORKED, NOW PUSH THE FIRE      BUTTON"
2400  FOR X = 1 TO 1000: IF  PEEK ( - 16287) < 128 THEN 2700: REM    BUTT
     ON 2
2500  NEXT X
2600  GOTO 3800
2700  HOME : POKE  - 16293,0: REM   READ VERTICAL BUTTONS
2800  HOME : PRINT "NOW PRESS THE JOYSTICK UP": FOR X = 1 TO 1000
2900  IF  PEEK ( - 16286) < 128 THEN 3200
3000  NEXT
3100  GOTO 3800
3200  HOME : PRINT "NOW TILT THE JOYSTICK DOWN": FOR X = 1 TO 1000: IF  PEEK
     ( - 16285) < 128 THEN  POKE  - 16294,0: RETURN : REM    END OF LOOP.
      TURN AN0 BACK OFF
3300  NEXT
3400  GOTO 3800: REM   LOOP ENDED WITH NO BUTTONS DETECTED
3500  HOME : PRINT "NEXT PRESS THE BACK SWITCH TO THE LEFT  AND WE'LL REP
     EAT THE TEST ON THE OTHER  JOYSTICK"
3600  GOSUB 1300: RETURN
3700  END
3800  HOME : PRINT  CHR$ (7);"NOTHING HAPPENED. CHECK THE SWITCH      SET
     TINGS AND CONNECTIONS THEN PRESS     SPACE TO START OVER": GET A$: RUN
```



```
3900  REM  ROUTINE TO READ BOTH ATARI JOYSTICKS BY TOGGLING ANNUNCIATOR Z
     ERO
4000  HOME : PRINT "NOW WE'LL TEST THE MIDDLE POSITION OF   THE REAR SWIT
     CH": PRINT "PRESS THE SPACE  BAR  WHEN YOU HAVE IT  SET"
4100  POKE  - 16296,0:JT$ = "LEFT": GOSUB 1300: REM   TEST LEFT STICK
4200  HOME : PRINT "NOW TRY THE RIGHT STICK"
4300  POKE  - 16295,0:JT$ = "RIGHT": GOSUB 1300: REM  TEST RIGHT STICK
4400  HOME : PRINT "NOW WE'LL TEST THE PADDLE READINGS.     PUT THE REAR
     SWITCH TO THE RIGHT AND    THE FRONT SWITCH TO THE REAR"
4500  PRINT "PRESS THE SPACE BAR WHEN READY";: GET A$
4600 Y = 0:BT = ( - 16287): GOSUB 5300: PRINT  CHR$ (7);:Y = 1:BT = ( - 1
     6286): GOSUB 5300: PRINT  CHR$ (7);: REM  Y IS THE PADDLE TO READ AN
     D BT IS THE BUTTON TO READ
4700  HOME : PRINT "NOW MOVE THE REAR SWITCH LEFT TO TEST   THE OTHER PAD
     DLES. PRESS SPACE WHEN     READY TO CONTINUE";: GET A$:Y = 0:BT = ( -
     16287): GOSUB 5300: PRINT  CHR$ (7);:Y = 1:BT = ( - 16286): GOSUB 53
     00: PRINT  CHR$ (7)
4800  HOME : PRINT "NEXT WE TEST THE MIDDLE POSITION OF THE REAR SWITCH.
     PRESS THE SPACE BAR WHEN   READY TO PROCEED";: GET A$
4900 Y = 0:BT = ( - 16287): GOSUB 5300: REM  READ PADDLE 0 AND BUTTON0
5000 Y = 1:BT = ( - 16286): GOSUB 5300: REM  READ PADLE 1 AND BUTTON1
5100 Y = 2:BT = ( - 16287): POKE  - 16295,0: GOSUB 5300: REM  READ PADDLE
     2 AND BUTTON0
5200 Y = 3:BT = ( - 16286): GOSUB 5300: REM  READ PADDLE3 AND BUTTON1
5250  RETURN
5300  REM  PADDLE AND BUTTON TEST FOLLOW
5400  HOME
5500  PRINT "TWIST THE KNOB ON PADDLE "Y" LEFT AND     RIGHT QUICKLY"
5600  FOR X = 1 TO 1000
5700  IF  PDL (Y) < 100 THEN 6000
5800  NEXT X
5900  GOTO 3800
6000  IF  PDL (Y) > 100 THEN 6300
6100  NEXT X
6200  GOTO 3800
6300  GOSUB 6400: RETURN : REM  GO TEST BUTTON
6400  PRINT  CHR$ (7);: HOME : PRINT "OK, NOW PUSH THE BUTTON": FOR X = 1
     TO 1000: IF  PEEK (BT) > 128 THEN  RETURN
6500  NEXT X
6600  GOTO 3800: REM  NOTHING DETECTED GO TO THE ERROR ROUTINE
```

Appendix B: Pascal Demo Program

```
PROGRAM JOYDEMO;

USES APPLESTUFF;

CONST   CLEARSCREEN = 12; (*CONTROL CHARACTER TO CLEAR APPLE SCREEN*)
        BELL = 7;         (*CONTROL CHARACTER TO SOUND BELL*)
        FOREVER = FALSE;  (*A LONG TIME*)

VAR CH:CHAR;
    X,Y,I:INTEGER;

PROCEDURE PAINTSCREEN;
(*PUT MENU ON SCREEN*)
BEGIN
  WRITELN(CHR(CLEARSCREEN),'apple mode':25);
  WRITELN;
  WRITELN('paddle Ø:  Ø button Ø:OFF button 3:OFF');
  WRITELN('paddle 1:  Ø button 1:OFF button 4:OFF');
  WRITELN('paddle 2:  Ø button 2:OFF button 5:OFF');
  WRITELN('paddle 3:  Ø');
  WRITELN;
  WRITELN;
  WRITELN('atari mode':25);
  WRITELN;
  WRITELN('    f u                    f u');
  WRITELN('    l   r                  l   r');
  WRITELN('      d                      d');
  WRITELN('MENU':22);
  WRITELN('apple/atari  controller set switch');
  WRITELN(' 1)Apple             left');
  WRITELN(' 2)Apple             right');
  WRITELN(' 3)Apple             middle');
  WRITELN(' 4)Atari             left');
  WRITELN(' 5)Atari             right');
  WRITELN(' 6)Atari             middle');
  WRITELN(' 7)end the demonstration');
  WRITELN('Set JOYPORT switches and then make');
  WRITE('your choice.');
  GOTOXY(44,Ø);
END;(*PAINTSCREEN*)

PROCEDURE PADDLESCANN;
(*READ THE PADDLES*)
BEGIN
  FOR I:=Ø TO 3 DO
  BEGIN
    GOTOXY(9,2+I);
    WRITE(PADDLE(I):3);
  END;
END;(*PADDLESCANN*)

PROCEDURE BUTTONSCANN(BSET:INTEGER);
(*READ THE PADDLE BUTTONS*)
BEGIN
  IF BSET=1 THEN X:=22 ELSE X:=35;
  FOR I:=Ø TO 1 DO
```



```
  BEGIN
    IF BUTTON(I) THEN
    BEGIN
      GOTOXY(X,2+I);
      WRITE(' ON');
      REPEAT
      UNTIL NOT(BUTTON(I));
      GOTOXY(X,2+I);
      WRITE('OFF');
    END;
  END;
END;(*BUTTONSCANN*)

PROCEDURE APLMODE1;
(*READ JOYPORT WITH CONTROLLER SELECT SWITCH IN LEFT OR RIGHT POSITION*)
BEGIN
  GOTOXY(Ø,1);
  WRITE('==========':25);
  REPEAT
    PADDLESCANN;
    BUTTONSCANN(1);
    IF KEYPRESS THEN
    BEGIN
      READ(KEYBOARD,CH);
      IF CH IN ['1'..'7'] THEN
      BEGIN
        GOTOXY(Ø,1);
        WRITE('          ':25);
        EXIT(APLMODE1);
      END ELSE WRITE(CHR(BELL));
    END;
  UNTIL FOREVER;
END;(*APLMODE1*)

PROCEDURE APLMODE2;
(*READ JOYPORT WITH CONTROLLER SELECT SWITCH IN MIDDLE POSITION. USE
  ANNUNCIATOR 1 TO SELECT BETWEEN THE PUSH BUTTONS ON THE LEFT & RIGHT
  OF BUTTONS
*)
BEGIN
  GOTOXY(Ø,1);
  WRITE('==========':25);
  REPEAT
    PADDLESCANN;
    TTLOUT(Ø,FALSE);(*TURN OFF ANNUNCIATOR 1*)
    BUTTONSCANN(1);
    TTLOUT(Ø,TRUE);(*TURN ON ANNUNCIATOR 1*)
    BUTTONSCANN(2);
    IF KEYPRESS THEN
    BEGIN
```

```
      READ(KEYBOARD,CH);
      IF CH IN ['1'..'7'] THEN
      BEGIN
        GOTOXY(Ø,1);
        WRITE('          ':25);
        EXIT(APLMODE2);
      END ELSE WRITE(CHR(BELL));
    END;
  UNTIL FOREVER;
END;(*APLMODE2*)

PROCEDURE HSCANN(JOYSTICK:INTEGER);
(*SCANN THE HORIZONTAL BUTTONS OF THE ATARI*)
BEGIN
  IF JOYSTICK=1 THEN X:=4 ELSE X:=26;
  TTLOUT(1,FALSE);(*SELECT THE HORIZONTAL BUTTONS*)
  GOTOXY(X,11);
  IF BUTTON(1) THEN WRITE('L')
    ELSE WRITE('l');
  GOTOXY(X+4,11);
  IF BUTTON(2) THEN WRITE('R')
    ELSE WRITE('r');
END;(*HSCANN*)

PROCEDURE VSCANN(JOYSTICK:INTEGER);
(*SCANN THE VERTICLE KEYS OF THE JOYSTICK*)
BEGIN
  IF JOYSTICK=1 THEN X:=6 ELSE X:=28;
  TTLOUT(1,TRUE);(*SELECT THE HORIZONTAL BUTTONS*)
  GOTOXY(X,1Ø);
  IF BUTTON(1) THEN WRITE('U')
    ELSE WRITE('u');
  GOTOXY(X,12);
  IF BUTTON(2) THEN WRITE('D')
    ELSE WRITE('d');
END;(*HSCANN*)

PROCEDURE FIRESCANN(JOYSTICK:INTEGER);
(*SCANN THE JOYSTICK FIRE BUTTON*)
BEGIN
  IF JOYSTICK=1 THEN X:=4 ELSE X:=26;
  GOTOXY(X,1Ø);
  IF BUTTON(Ø) THEN WRITE('F')
    ELSE WRITE('f');
END;(*FIRESCANN*)

PROCEDURE ATARI1;
(*SCANN THE JOYSTICK WHEN CONTROLLER SELECT IS LEFT OR RIGHT*)
BEGIN
  GOTOXY(Ø,9);
  WRITE('==========':25);
  REPEAT
```



```
    HSCANN(1);
    VSCANN(1);
    FIRESCANN(1);
    IF KEYPRESS THEN
    BEGIN
      READ(KEYBOARD,CH);
      IF CH IN ['1'..'7'] THEN
      BEGIN
        GOTOXY(Ø,9);
        WRITE('          ':25);
        EXIT(ATARI1);
      END ELSE WRITE(CHR(BELL));
    END;
  UNTIL FOREVER;
END;(*ATARI1*)

PROCEDURE ATARI2;
(*SCANN THE JOYSTICK WHEN CONTROLLER SELECT IS IN CENTER*)
BEGIN
  GOTOXY(Ø,9);
  WRITE('==========':25);
  REPEAT
    TTLOUT(Ø,TRUE);(*SELECT THE LEFT JOYSTICK*)
    HSCANN(1);
    VSCANN(1);
    FIRESCANN(1);
    TTLOUT(Ø,FALSE);(*SELECT THE RIGHT JOYSTICK*)
    HSCANN(2);
    VSCANN(2);
    FIRESCANN(2);
    IF KEYPRESS THEN
    BEGIN
      READ(KEYBOARD,CH);
      IF CH IN ['1'..'7'] THEN
      BEGIN
        GOTOXY(Ø,9);
        WRITE('          ':25);
        EXIT(ATARI2);
      END ELSE WRITE(CHR(BELL));
    END;
  UNTIL FOREVER;
END;(*ATARI1*)
```

```
BEGIN (*JOYDEMO*)
  PAINTSCREEN;
  REPEAT
    READ(KEYBOARD,CH);
  UNTIL CH IN ['1'..'7'];
  REPEAT
    CASE CH OF
      '1':BEGIN
            GOTOXY(0,15);WRITE('*');
            APLMODE1;
          END;
      '2':BEGIN
            GOTOXY(0,16);WRITE('*');
            APLMODE1;
          END;
      '3':BEGIN
            GOTOXY(0,17);WRITE('*');
            APLMODE2;
          END;
      '4':BEGIN
            GOTOXY(0,18);WRITE('*');
            ATARI1;
          END;
      '5':BEGIN
            GOTOXY(0,19);WRITE('*');
            ATARI1;
          END;
      '6':BEGIN
            GOTOXY(0,20);WRITE('*');
            ATARI2;
          END;
      '7':BEGIN
            WRITELN(CHR(CLEARSCREEN));
            WRITELN('THAT''S ALL FOLKS');
            EXIT(JOYDEMO);
          END;
    END;(*CASE*)
    FOR I:=15 TO 20 DO
    BEGIN
      GOTOXY(0,I);
      WRITE(' ');
    END;
  UNTIL FOREVER;
END.(*JOYDEMO*)
```


Appendix C: Pascal Graphics Editor Update

If you purchased the Pascal Graphics Editor, several additions were made to it which will allow you to use the JOYPORT to directly draw on the screen. These additions are included in the backup copy of PGE supplied on the UTILITIES DISK. Included in the revised version are cursor control using an Atari Joystick and the B(rush command in the T(ext level.

To use the JOYPORT interface to PGE, first set the ATARI/APPLE switch to the ATARI position. Make sure you have an Atari-type controller plugged into the JOYPORT and that the CONTROLLER SELECT SWITCH is pushed over to the side the joystick is plugged into. Boot the backup copy of PGE supplied on the front of the UTILITIES DISK. Proceed as usual, but once the PGE command line appears type "J" to toggle on the JOYPORT interface. If you wish to disable the JOYPORT interface later, return to the PGE command level and press "J". Each successive time you press "J" the JOYPORT interface will be enabled or disabled. The new capabilities available are described below:

S(ketch Level:

The Atari-type controller can be used to position the cursor on the screen by moving the stick in the direction you wish to move the cursor. Pressing the fire button will cause a line to be drawn from the last point plotted to the current cursor position. All other keyboard commands remain in effect as before.

T(ext Level:

The text cursor can now be positioned on the screen using the joystick. In addition, a brush mode has been added to the T(ext level.

The B(rush allows you to continuously draw text characters on the screen using either the keyboard or the joystick. To use this feature, press "ESC" followed by "B". The prompt line at the bottom of the screen will then look like:

→ *LETTER:*

Enter the letter you want to continuously draw on the screen. Next the prompt will ask for the brush height. Enter the height of the brush you wish to draw with (must be between 1 and the current text height).

When you move the cursor with the joystick or the keyboard it will leave a trail of letters behind it that are the defined letter width and the height you entered above. If you wish to position the cursor to a location on the screen without leaving a trail of letters, press the fire button. The cursor will now move without leaving a trail. When you are ready to paint with the brush again, press the fire button again.

You can use the text D(efine mode to create new brush characters. Try using different combinations of dots and letter widths to create thin and fat brushes. If you fill in every other dot across the width of a character cell and set the width to an even number the brush will "paint" in color.

To quit the brush mode, type "ESC" followed by "B". When you are prompted for the letter to use respond by pressing the space bar. This will return you to the normal text mode.

M(ove Level:

Use the joystick to position the cursor. The fire button will set a point at the edge of the area you wish to move (it functions the same as the "." key does).

D(irectory, L(oad and W(rite are not affected by the JOYPORT interface.